OBJECT COMPUTING
HOME TO MICRONAUT®

CASE STUDY: FROM MONOLITH TO MICROSERVICES WITH MICRONAUT®

A large, consumer-facing IoT device network that relied upon a monolithic architecture was struggling to keep pace with increasing demand. In response, the team built a lightweight microservices system using the Micronaut framework, which provided the smooth learning curve, extensibility, speed, testing ease, and scalability necessary to better serve its growing customer base.

THE CLIENT

SmartThings, a subsidiary of Samsung Electronics, provides consumers at-home IoT experiences through a range of sensors, smart devices, and digital apps. SmartThings customers have the ability to control, automate, and monitor household fixtures, such as lights, locks and security systems, electrical outlets, and more, via their mobile devices.

THE CHALLENGE

The key benchmarks for measuring success using SmartThings' "smart" products are speed and accuracy. Homeowners who invest in smart technology expect immediate and consistent responses to both scheduled and spontaneous commands.

SmartThings' existing monolithic application controlled approximately 550 services from a central server. The compute power necessary to manage this load was not only unwieldy, it resulted in unacceptable startup times during peak usage.

Additionally, extensive manual testing was required for each software change, and a single update test could take between 45 to 90 minutes for each service. This made continuous deployment difficult and resulted in taxing development cycles.

Finally, the company's growth exhausted compute resources and ability to scale; the more commands executed by SmartThings users, the greater its stress on the application. To further complicate matters, unpredictable surges in user behavior caused periods of reduced performance.

THE OBJECTIVES

SmartThings set out with three key objectives:

1.The SmartThings Hub and its mobile app required a new infrastructure that would deliver performance of critical operations at sub-second speeds. To deliver the speed and scale expected by its customer base, SmartThings set a 400-millisecond max on response time. (This time limit represents the threshold of human perception, meaning any span of time longer than 400 milliseconds is likely to be noticeable as a "delay" to the user.)

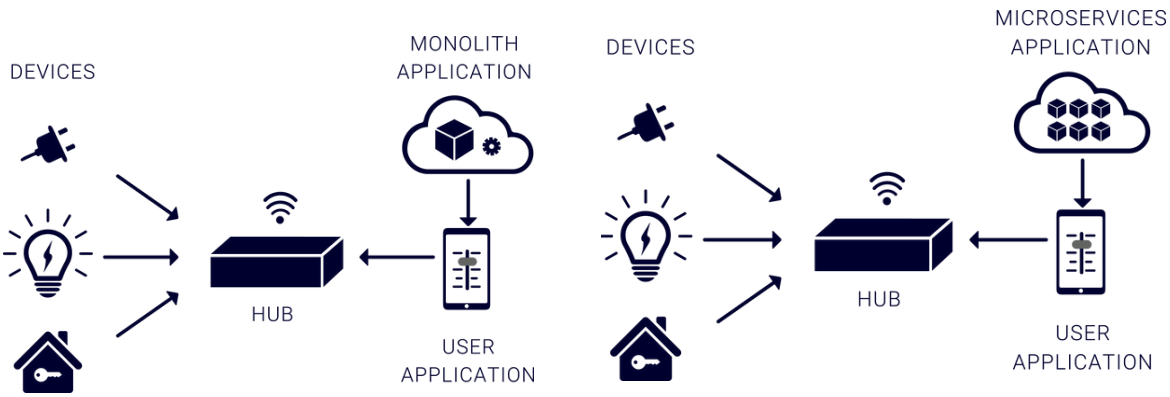
2. Any new feature implemented needed to be extensible into the distant future, and the company's engineers sought to reduce as much boilerplate code as possible to accelerate development cycles. Thus, only solutions that empowered the SmartThings engineering team to rapidly develop, test, and release new software features were considered.

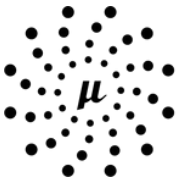
3. Finally, the new system had to be scalable to both accommodate the growing number of potential users and to address load during peak use times.

THE SOLUTION

SmartThings recognized that it needed to transition away from its legacy monolith architecture to a more flexible microservice-based system.

A microservice architecture allows for nimble development and delivery of a wide range of interoperable services that may be modified and deployed independently. Such a solution clearly offered significant benefits to both the internal development team and the company's client base.





OBJECT COMPUTING
HOME TO MICRONAUT®

THE SOLUTION (cont.)

Once SmartThings leaders decided to transition their massive and complex existing system to a lightweight, interconnected infrastructure, they faced the challenge of building and migrating to their new platform as efficiently as possible.

They sought a framework that fulfilled all of the following requirements:

- It had to be intuitive and easy for their development team to learn and use.
- It had to be robust to handle a vast (and growing) number of services without slowing down or exceeding compute resources.
- It had to meet the speed, ease-of-testing, and scalability requirements outlined above.

After attempting to build a solution using Ratpack, SmartThings' technical leadership turned to the Micronaut framework, an open source JVM framework specifically designed for building scalable, highly performant microservices and serverless applications.

The Micronaut framework fulfilled all of SmartThings' requirements:

✔ A smooth learning curve

✔ Robust, extensible framework that enables speed and scalability

✔ Easy-to-test code

THE OUTCOMES

By leveraging the Micronaut framework's cloud-native features and unique sensible defaults, SmartThings has achieved the following business benefits:

- Consolidated servers
- Increased workload efficiency in the cloud
- Shorter development cycles
- Reduced and better-controlled costs

"We initially selected Ratpack to transition our monolith application to a microservices architecture, but we found that the learning curve was too steep, and we were not as productive with Ratpack as we were with Micronaut."

- SmartThings Engineer

5494 DEC 2020